A Robust Hough Transform Based on Validity

Jongwoo Kim, University of Missouri, Columbia, MO 65211
Raghu Krishnapuram, Colorado School of Mines, Golden, CO 80401
kimjw@ece.missouri.edu, rkrishna@mines.edu

Abstract

In this paper, we propose a Robust Hough Transform (RHT) which addresses many of the problems associated with the conventional Hough Transform (HT). The bin-splitting problem is solved by the use of robust clustering for peak detection, the accuracy problem is solved by means of an analog Hough space, the bias problem is solved by the multiple-point method combined with random sampling, and the spurious peak problem is solved by the use of a validity measure. We present experimental results of the proposed RHT on real images.

1. Introduction

1.1 Background

The Hough Transform (HT) is one of the most widely used algorithms for the detection of straight lines and curves in computer vision [1-3]. It uses an accumulator which discretely approximates the parameter space or Hough Space (HS). Typically, each point in image space votes along a curve in parameter space and the votes are accumulated in HS. The HT is a very popular parameter estimation and shape detection technique due to its "robustness" in the presence of noise and due to its implementational simplicity. However, the HT technique suffers from several major drawbacks which prevent its application to complex data sets. In this paper, we address these drawbacks and propose some solutions.

1.2 Problems Associated with the HT

Several problems with the conventional HT and the proposed solutions are listed below. (i) The peaks in HS represent the parameters of the lines (or curves) in the image. When edge points are scattered in the image (which is partly due to the discrete nature of image space), or when the edges are thick, the peak corresponding to a curve in image space will be distributed among several neighboring bins in HS. This makes peak detection difficult, and gives rise to the so-called "bin-splitting" problem. The bin-splitting problem is also partly due to the discrete nature of HS. We propose the use of a robust clustering algorithm in HS to solve this problem. (ii) The discrete nature of HS causes the parameter estimation to be inaccurate since the accuracy of the estimates depends on the resolution of HS. Increasing the resolution worsens the bin-splitting problem. We use an analog HS to solve this problem. (iii) In the HT, a point in image space votes for parameters corresponding to all curves passing through it. Therefore, all votes, except for the one corresponding to the actual curve, are wrong. This creates a serious "bias" problem in most accumulator cells in HS. The bias, which essentially acts as noise, makes the detection of peaks difficult. We propose a voting scheme based on the multiple-point method combined with random sampling in a local window to mitigate this problem. (iv) Noise points in image space can give rise to false peaks in HS by accidentally "lining up" to form a curve. As a result, false peaks can occur often in the HT. This problem is also solved by the use of validity-based voting. (v) In the case of the straight-line HT, when there are several (collinear) straight lines with the same equation but different starting and ending points, the conventional HT cannot distinguish them. We solve this problem by using the mid point information of straight lines in a 4-D HS .

There are several variations of the HT [4-8,11] in the literature that address one or more of the above-mentioned issues. Our approach is an attempt at finding a solution that addresses all of the issues.

2. Robust HT with Validity-Based Voting

The proposed algorithm is composed of four parts. The first part is a preprocessing step to compute an edge image from an input gray-level image. We do not use a thinning step because the proposed HT is able to handle thick edges. The second part consists of computing an analog Hough transform using robust estimators and a validity measure. The third part uses a clustering algorithm to estimate the number of clusters (peaks) in the analog Hough Space (HS). The clustering algorithm is an unsupervised possibilistic clustering algorithm based on the least trimmed squares estimator of robust statistics [9]. The fourth part is a post-processing algorithm to remove spurious clusters (peaks). We now explain each step in more detail.

2.1 Computing the Edge Image

The Sobel operator was used in our experiments. After estimating the magnitude of an input image using the

0-7803-4863-X/98 $10.00©1998 IEEE

Sobel operator, we use a threshold to obtain a binary image.

2.2 Robust Analog HT

To estimate the parameters of a curve with p parameters, we need p points from the curve, i.e., we need a p-subset of the data set. We compute the robust HT by randomly picking several p-subsets in a window. Thus, the size of the window very much affects the result of the estimation. In our experiments, we use a window of size $W\times W$=21×21 under the assumption that the maximum width of the lines in the images is less than $(W-1)/2$=10. We shift the window 5 pixels at a time in raster scan order to compute the HT of the whole input image. If ε is the fraction of the noise points, the probability P that at least one out of the M selected p-subsets is good, can be expressed as

$$P = 1-(1-(1-\varepsilon)^p)^M.$$

We can choose the value of M depending on the desired value of P. When we have thick lines in the window, all the points in the window except those corresponding to the skeleton are essentially noise. If there are N_W edge pixels in the image, then by assuming that there will be two lines of length $(W-1)/2$ in each window, we estimate the line width W_L as $N_W/(W-1)$, and the noise fraction ε as $(W_L - 1)/W_L$. We use P=0.99 and p=2 in our experiments, and restrict $M\leq 100$ for each window to reduce the computational burden.

In the case of a straight line, p=2, for the normal parametrization of HS, the angle θ_i and the perpendicular distance ρ_i can be computed from two points

$$\mathbf{x}_{i1} = (x_{i1}, y_{i1}) \text{ and } \mathbf{x}_{i2} = (x_{i2}, y_{i2}) \quad (1)$$

as follows:

$$\theta_i = \tan^{-1}\left(\frac{x_{i1}-x_{i2}}{y_{i2}-y_{i1}}\right) \text{ and}$$

$$\rho_i = x_{i1}\cos\theta_i + y_{i1}\sin\theta_i. \quad (2)$$

Once the parameters (θ_i, ρ_i) of the line corresponding to a pair of points have been estimated, we identity the subset X_{Si} of (edge) image points associated with (θ_i, ρ_i) from the total data set in six steps. (The steps are described below.) Then the validity of (θ_i, ρ_i) is computed based on X_{Si}. This is repeated for all M 2-subsets that we pick in the window.

Step 1: Initially X_{Si} is considered to be the set X of all edge pixels in the image. Then, the Histogram of Residuals (HR) is computed of all points $\mathbf{x}_j$ in X_{Si} for which $-W\leq r_j \leq W$. The residual r_j of a point $\mathbf{x}_j = (x_j, y_j)$ is given by $r_j = \text{Integer}(x_j\cos\theta_i + y_j\sin\theta_i - \rho_i)$. We use a bin width of 1 in HR, corresponding to 1 pixel resolution. The histogram is smoothed and the residual corresponding to the mode of the smoothed histogram is considered to be the center of HR. Then all points in the image that are not within $1.5\times(W-1)/2$ on each side of the center are trimmed (i. e., removed from X_{Si}).

Step 2: In this step, the points remaining in X_{Si} are projected onto the straight line (θ_i, ρ_i) and accumulated to create Projection Space (PS) corresponding to (θ_i, ρ_i). Bin size = 1 is used in this PS, again representing 1 pixel width. Distance dI_j associated with each non-empty bin j in PS is computed as $dI_j = (n-j)$, where n is the index of the nearest non-empty bin in the positive direction from bin j. The inter-point distance associated with the line is computed as

$$I = C\times \underset{j}{med}(dI_j), \quad (3)$$

where C=2 in our experiments. We then perform 1-D "blob coloring" (component labeling) on PS. For the purposes of blob coloring, all non-empty bins are treated as "object pixels". When the distance between two bins is less than I, the two bins are considered connected. After blob coloring, we identify the blobs (components) which are not connected to the component containing the projections of the two sampled points in (1), and remove from X_{Si} points that project to the these unconnected blobs.

Step 3: In this step, we repeat the same procedure as in Step 1 to construct a HR and trim the points which are not within $(W-1)/2$ on each side of the center of the HR. The boundaries (cut-off points) $(-B, B)$ on both sides are computed using

$$B = \min\left(C\times \underset{j}{med}(|RA_{neg}[j]|), C\times \underset{j}{med}(|RA_{pos}[j]|)\right),$$

where $RA_{neg}[j]$ and $RA_{pos}[j]$ are the histogram arrays of negative and positive residuals of the points in X_{Si}. When the residual of a point is zero, it is accumulated in both RA_{neg} and RA_{pos}. Before estimating boundaries of each side, we make the histograms of $RA_{neg}[j]$ and $RA_{pos}[j]$ symmetric about zero, i.e., after estimating the histograms of $RA_{neg}[j]$ and $RA_{pos}[j]$, we flip the histograms about zero and add it to the original histograms. We use a value of 3 for C, corresponding to a $\pm 3\sigma$ cut-off for Gaussian distributions.

Step 4: This step is a repetition of Step 2.

Step 5: This is the final step of trimming. In this step, the maximum absolute residual of each cell in projection space on each side (positive and negative) is estimated. Let $MPS_{neg}[j]$ and $MPS_{pos}[j]$ denote the maximum absolute residuals on the negative and positive sides in the j-th bin of projection space PS. Then, final boundaries of negative and positive sides are estimated as

$$FB_{neg} = \underset{j}{med}(MPS_{neg}[j]) - C\times \underset{j}{MAD}(MPS_{neg}[j])$$

and

$$FB_{pos} = \underset{j}{med}(MPS_{pos}[j]) + C \times \underset{j}{MAD}(MPS_{pos}[j]).$$

In the above equations, MAD is the median of absolute deviation, which is a robust estimate of the standard deviation. C=2 is used in our experiments, corresponding to $\pm 2\sigma$ cut-off points for Gaussian distributions.

Step 6 : Using the remaining data in X_{Si} and the parameter estimates (θ_i,ρ_i), we can estimate the starting point $\mathbf{s}_i = (x_{si}, y_{si})$, and ending point $\mathbf{e}_i = (x_{ei}, y_{ei})$ of projection space PS, as well as the center $\mathbf{c}_i = (x_{ci}, y_{ci})$, which is the average of $\mathbf{s}_i$ and $\mathbf{e}_i$.

After estimating the final X_{Si}, we compute the validity of the estimated line (θ_i,ρ_i) using the formula:

$$V_i = \frac{N}{I_D \cdot (1+\alpha B_f) \cdot (1+\beta(\delta / L))(1+Res_{avg})}, \quad (4)$$

where N is the number of points in X_{Si}, $\delta = FB_{pos} - FB_{neg} + 1$ is the width of the straight line, and Res_{avg} is the average of squared residuals of the points in X_{Si}. The formulas for the remaining quantities are given below.

The length L is given by

$$L = dL\sqrt{1+\tan\theta_o}, \quad (5)$$

where $dL = dL_{start} - dL_{end} + 1$, dL_{start} is the smallest index i for which $PS[i] > 0$, dL_{end} is the largest index i which is $PS[i] > 0$ in the final projection space PS corresponding to X_{Si}, and

$$\theta_o = \begin{cases} \theta_i & if\ 0 \le \theta_i < \pi/4 \\ \pi/2 - \theta_i & if\ \pi/4 \le \theta_i < \pi/2 \\ \theta_i - \pi/2 & if\ \pi/2 \le \theta_i < 3\pi/4 \\ \pi - \theta_i & if\ 3\pi/4 \le \theta_i < \pi \end{cases}. \quad (6)$$

Tangent term in (5) is used to compensate for digitization effects. The average inter point distance I is estimated by the following equation:

$$I = \frac{dL-1}{PSGZ-1}\sqrt{1+\tan\theta_o}, \quad (7)$$

where $PSGZ$ is the number of projection bins that have values greater than zero, and θ_o is as defined in (6). The constant term β has the smallest value when the width of the straight line is 1 and has the largest value when the $L=\delta$. It is given by

$$\beta = \frac{L^h}{(L-\delta)^h + 1}.$$

The power h is used for "sharpening", and h=6 is used in our experiments. $B_f \in [0,1]$ is the balancing factor to ameliorate the bin-splitting problem. It is given by,

$$B_f = \frac{\sum_{i(i=i+I)} \left| \left| \sum_{j=i}^{i+I-1} SPS_{neg}[j] \right| - \left| \sum_{j=i}^{i+I-1} SPS_{pos}[j] \right| \right|}{\sum_j \left(|SPS_{neg}[j]| + |SPS_{pos}[j]| \right)},$$

where SPS_{neg} and SPS_{pos} are the accumulators of the squares of the negative and positive residuals of points in X_{Si} in projection space PS, and I is the inter-point distance in (7). When the points are "balanced" on both sides of the line, the numerator term is roughly 0, and B_f has a small value. When the deviations are not balanced, B_f has large value and the validity is decreased. The constant α is used to magnify the role of B_f in the validity measure. Let the variances of SPS_{neg} and SPS_{pos} be σ^2_{neg} and σ^2_{pos}. Then, α is estimated as $\alpha = 1 + (\sigma^2_{neg} + \sigma^2_{pos})/2$.

After estimating the validity V_i associated with X_{Si}, we record V_i, and $\mathbf{q}_i = (\theta_i, \rho_i, x_{ci}, y_{ci})$ in an analog HS. The HS needs to be thresholded depending on the minimum acceptable validity for a straight line, which depends on the application. We recommend a threshold of 3, since this could find all well-formed lines containing more than 3 pixels. A higher threshold may be needed in the case of noisy images, to avoid detecting too many small spurious lines.

2.3 Robust Clustering in Analog Hough Space

We use the FCM algorithm followed by an unsupervised possibilistic [10] clustering algorithm to find the peaks in the four-dimensional HS. This procedure, which we refer to as the Unsupervised Possibilistic Trimmed C Prototypes (UPTCP) algorithm, is summarized below.

The Unsupervised Possibilistic Trimmed C Prototypes (UPTCP) Algorithm

Let $\mathbf{p}_i = (\theta_i, \rho_i, x_{ci}, y_{ci})$ denote the cluster prototypes for i=1, ..., C;
Let V_j denote the validity of the four-dimensional point $\mathbf{q}_j$, j=1, ..., N computed using (4);
Fix C, $2 \le C \le N$, fix $m \in (1,\infty)$, fix step size for trimming Δp;
Initialize the C clusters using the FCM algorithm;
Set p=100 {p = retention percentage = $(100 \times P)/N$};
Repeat
  Estimate the resolution parameter η_i for the clusters (see below);
  Repeat
    Compute the harmonic distances

$$h_j^2 = \frac{1}{C}\left(\sum_{i=1}^{C} d_{ij}^{2/(1-m)}\right)^{1-m} \quad \text{for } j = 1, \cdots, N,$$

    where d_{ij}^2 is the distance of $\mathbf{q}_j$ from $\mathbf{p}_i$;
    Sort h_j^2 in ascending order to create $h_{j:N}^2$, $j = 1, \cdots, N$ and trim data by keeping only points corresponding to the smallest $P = N \times p/100$ harmonic distances;
  Compute the memberships

$$u_{ij:N}^{(l)} = \frac{d_{ij:N}^{2/(1-m)}}{\sum_{k=1}^{C} d_{kj:N}^{2/(1-m)}};$$

Estimate C possibilistic cluster prototypes

$$\mathbf{p}_i = \sum_{j=1}^{P} u_{ij:N}^{m} V_{j:N} \mathbf{q}_{j:N} \Big/ \sum_{j=1}^{P} u_{ij:N}^{m} V_{j:N};$$

Until change in prototype parameters is very small;

If $\| \mathbf{p}_i - \mathbf{p}_j \| < \varepsilon$, merge cluster i and j;

$p = p - \Delta p$;

Until ($p \le 50$);

Crisply assign each point $\mathbf{q}_j$ in HS to a cluster;

Pick the point $\mathbf{q}_j$ with the highest validity in each cluster as the prototype of the cluster;

The above algorithm starts with an overspecified number of clusters and then merges clusters that are close. We use $\varepsilon = 0.001$ and $\Delta p = 5$ in our experiments. The resolution parameter η_i is estimated as follows. After initializing with FCM, each point $\mathbf{q}_i$ in HS is crisply assigned to a cluster. Then η_i is computed as $A \times \underset{k}{med}(ME_k)$, $i = 1, \ldots, C$, where $ME_i = \underset{j}{med}(d_{ij}^2)$. Normally A should be roughly 3 for Gaussian clusters. However, since the clusters in HS are not Gaussian, we choose a more pessimistic value, i.e., $A = 2 \times 3 = 6$.

2.4 Removing Spurious Peaks

The clustering algorithm used in the above section does not always perform perfectly. Isolated peaks which correspond to noise can sometimes survive until the end of the clustering stage. A post-processing algorithm is needed to remove the spurious peaks after clustering in analog HS. We now briefly explain this algorithm, which we refer to as the Spurious Peak Removal (SPR) algorithm. Let the validity associated with a point $\mathbf{x}_j \in X$ in the edge image be $V[j] = \max_{i:\mathbf{x}_j \in X_{Si}} V_i$, where V_i is the validity of the line (see (4)) corresponding to the subset $X_{Si} \subset X$ represented by the line (prototype) $\mathbf{p}_i$. In other words, if a point belongs to only one line, then the validity of the point is equal to the validity of the line. If the point belongs to multiple lines, then the validity of the point is equal to the validity of the line that has the highest validity. Now consider a subset X_{Vi} of X_{Si} that includes only those points that have validities equal to the validity V_i of X_{Si}, i.e., $X_{Vi} \subset X_{Si} = \left\{ \mathbf{x}_j \in X_{Si} \middle| V_j = V_i \right\}$. Let N_{Si} denote the cardinality of X_{Si}, and let N_{Vi} denote the cardinality of X_{Vi}. The SPR algorithm deletes the prototype $\mathbf{p}_i$ if $N_{Vi}/N_{Si} < \varepsilon$.

When a small straight line meets two long straight lines at both ends forming a "U", the validity of the small straight line will be smaller than those of the two long straight lines. Therefore, the data points at both ends of the small straight line will have higher validity when they are included in one of the subsets representing the two long straight lines. This can mean that small straight lines that join long lines can be discarded as spurious, since the ratio N_{Vi}/N_{Si} for such lines will be small. To mitigate this problem, we use ε=0.3, even though we should use ε=0.5 (corresponding to 50%) from a robust statistical point of view.

3. Experimental Results

We first present an example that compares the RHT with the conventional HT. We then show examples of the proposed RHT. Figure 2 shows the 3-D plot of the conventional HT of the image in Figure 1. Peaks with less than 3 votes are set to zero in this HS. There are a lot of spurious peaks all over HS and the peaks also have flat tops because of bin-splitting. Figure 3 shows the 3-D plot of the (θ,ρ) sub-space of the RHT for the same image. Points with validity less than 3 are trimmed in this HS as well. There is almost no noise in the HS of RHT and the peaks also show very sharp characteristics. Because of the enormous number of bias points it was very difficult to obtain a good result with the conventional HT even when UPTCP algorithm was used to detect the peaks. The best result was obtained when we thresholded the HS in Figure 2 at 40 and then applied the UPTCP algorithm. Figure 4 shows the final result. Figure 5 shows the result obtained by RHT followed by UPTCP. Only the original thresholding (at 3) was used in this case.

Figure 6(a) shows a 256×256 image with three objects. Figure 6(b) shows the corresponding edge image. Figures 6(c) and (d) show the results after UPTCP, and SPR are applied. The overspecified number of clusters in the UPTCP algorithm was 23. The algorithm misses two short straight lines. This is because more than two thick straight lines are attached to these two small straight lines, and they take away most of the points belonging to the short lines. A better result can be obtained by thinning the edge image.

Figure 7(a) shows a noisy version of the image in Figure 6(a) obtained by adding Gaussian noise. Figure 7(b) shows the edge image and Figures 7(c) and (d) show the result after the UPTCP algorithm and the SPR algorithm have been applied. As we can see in Figure 7(d), the result is quite good, though it could not detect two small straight lines.

4. Summary and Conclusions

In this paper we have presented a robust Hough transform that overcomes most of the drawbacks of the conventional Hough transform. Although we have illustrated the proposed technique using data sets containing straight lines, the underlying technique is quite

general, and can be easily extended for more complex shapes by defining a suitable validity measure.

The proposed RHT cannot be directly compared with the conventional HT because the conventional algorithm uses a discrete 2-D HS and the RHT algorithm uses an analog 4-D HS. The RHT gives a lot more information about the detected lines, for example the starting and ending points. It can also distinguish between collinear lines. The computational complexities of the two methods can be estimated as follows. In the case of an image with N edge points, the conventional HT needs $O(N \times N_\theta)$ computations, where N_θ is the number of discrete θ values used in the HT. The proposed RHT needs approximately $O(4 \times M^2(6N + 2M + 12\log_2 N))$ computations. We obtain this result as follows. When N edge points are regularly distributed in an $M \times M$ edge image, $M^2/25$ widows are selected since we shift windows 5 pixels in each direction. Approximately $N \times W^2/M^2$ points are in a window, and a maximum of 100 point pairs are sampled in a window in our experiments. For a pair points, we need $O(N)$ computations in Step 1, $O(N+M)$ computations in Step 2, $O(N + 4\log_2 N)$ computations in Step 3, $O(N+M)$ computations in Step 4, $O(N + 8\log_2 N)$ computations in Step 5, and $O(N)$ computations in Step 6. Hence, $O(6N + 2M + 12\log_2 N)$ computations are needed for a pair of edge points. Therefore, when $N \gg M$ and $N \gg 12\log_2 N$, the computation of the RHT algorithm is almost $O(M^2/N_\theta)$ times greater than that of the conventional 2-D HT. We are currently investigating methods to speed up the proposed RHT.

References

[1] P. V. C. Hough, Methods and Means for Recognizing Complex Patterns, U.S. Patent 3 069 654, 1962.

[2] R. O. Duda and P. E. Hart, "Use of the Hough Transform to detect lines and curves in pictures," *Communications of the ACM*, Vol. 15, pp. 11-15, 1972.

[3] J. Illingworth and J. Kittler, "A Survey of the Hough Transform," *Computer Vision, Graphics, and Image Processing,* Vol. 44, pp. 87-116, 1988.

[4] J. H. Han, L. T. Kóczy, and T. Posten, "Fuzzy Hough Transform," *Pattern Recognition Letters*, 15, pp. 649-658, 1994.

[5] B. Kamgar-Parsi, B. Kamgar-Parsi, and N. S. Netanyahu, "A Nonparametric Method for Fitting a Straight Line to a Noisy Image," *IEEE Transactions on PAMI*, Vol. 11, No. 9, pp. 998-1001, September, 1989.

[6] W. Lie, and Y. Chen, "Robust Line-Drawing Extraction for Polyhedra Using Weighted Polarized Hough Transform," *Pattern Recognition*, Vol. 23, pp. 261-274, 1990.

[7] N. Kiryati, Y. Eldar, and A. M. Bruckstein, "A Probabilistic Hough Transform," *Pattern Recognition*, Vol. 24, pp. 303-316, 1991.

[8] Mark C. K. Yang, Jong-Sen Lee, Cheng-Chang Lien, and Chung-Lin Huang, "Hough Transform Modified by Line Connectivity and Line Thickness," *IEEE Transaction on PAMI*, Vol. 19, No. 8, pp. 905-910, August, 1997.

[9] J. Kim, R. Krishnapuram, and R. N. Davé, "Application of the Least Trimmed Squares Technique to Prototype-Based Clustering," *Pattern Recognition Letters*, 17, pp. 633-641, 1996.

[10] R. N. Davé and R. Krishnapuram, "Robust Clustering Methods: A Unified View," *IEEE Transaction on Fuzzy Systems*, Vol. 5, No. 2, pp. 270-293, May, 1997.

[11] G. Stockman, "Object Recognition and Localization via POSE clustering", *Comp Vision, Graphics, and Image Processing*, Vol. 40, No. 3, Dec. 1987, pp. 361-387.

Acknowledgment:

Partial support of this work by a grant from the Office of Naval Research (N00014-96-1-0439) is gratefully acknowledged.

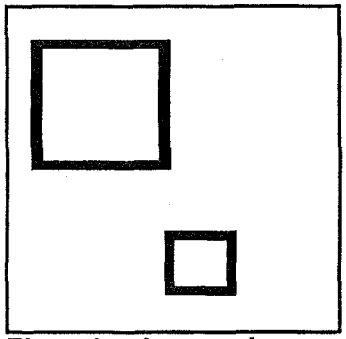


Fig. 1. Input image with thick edges.

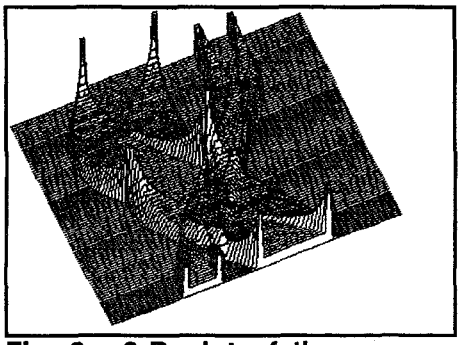


Fig. 2. 3-D plot of the conventional HT of the image in Fig. 1. The horizontal axis is ρ from –256 to 256 and vertical axis is θ from 0° to 179°.

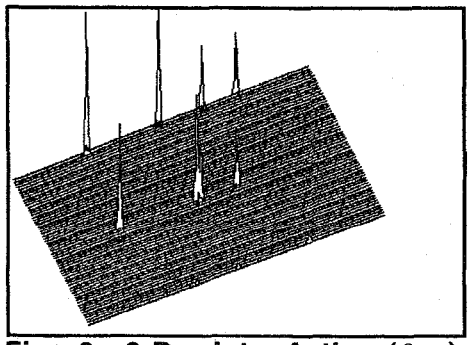


Fig. 3. 3-D plot of the (θ,ρ) Projection of proposed RHT of image in Fig. 1. The horizontal axis is ρ from –256 to 256 and vertical axis is θ from 0° to 179°.

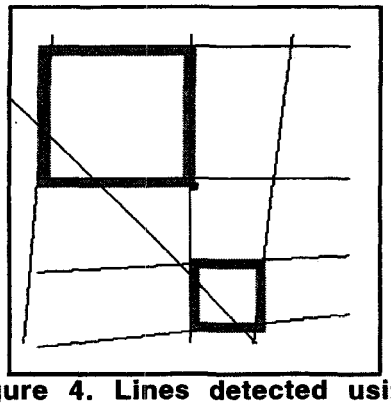


Figure 4. Lines detected using the conventional HT.

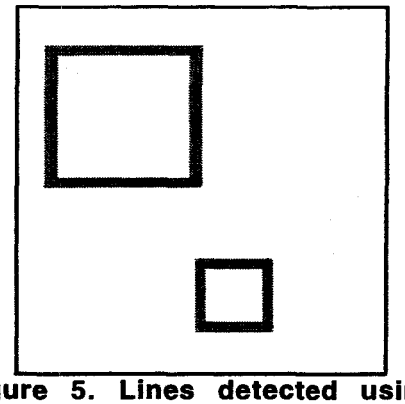
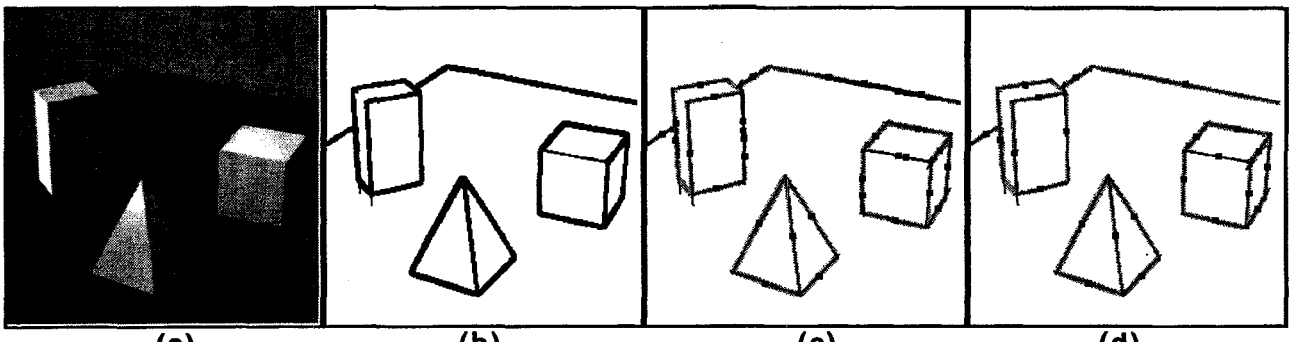
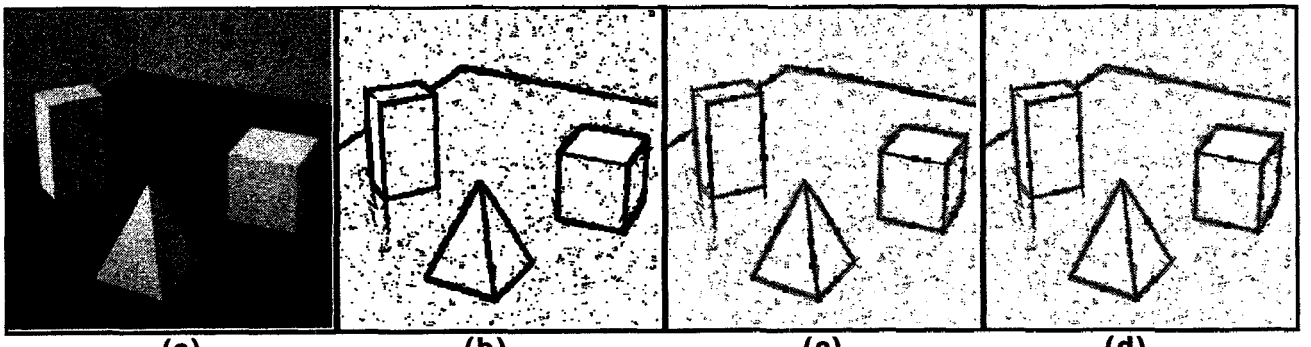


Figure 5. Lines detected using the proposed RHT.



(a) (b) (c) (d)

Figure 6. Result of the RHT on a complex image. (a) Input image. (b) Edge image obtained from the Sobel operation and thresholding. (c) Result of UPTCP. (d) Final result after SPR.



(a) (b) (c) (d)

Figure 7. Result of the RHT on a complex noisy image. (a) Input image. (b) Edge image obtained from the Sobel operation and thresholding. (c) Result of UPTCP. (d) Final result after SPR.